Mbstowcs

Ensure output buffer size is properly specified and large enough

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-26

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5482 bytes

<table>
<tr><td><b>Attack Category</b></td><td>• Malicious Input
• Denial of Service
• Privilege Exploitation</td></tr>
<tr><td><b>Vulnerability Category</b></td><td>• Multibyte Character
• Buffer Overflow
• No Null Termination</td></tr>
<tr><td><b>Software Context</b></td><td>• String Conversion MACROS</td></tr>
<tr><td><b>Location</b></td><td>• stdlib.h</td></tr>
<tr><td><b>Description</b></td><td>When using mbstowcs(), one must ensure that the output buffer is large enough and its size is correctly specified.
The mbstowcs() function converts a multibyte string src to a wide-character string. Multibyte strings can have a variable number of bytes per character, while wide-character strings are Unicode, with two bytes per character.
Problems can result if either (1) the result of the output buffer is specified incorrectly, permitting a buffer overflow to occur; or (2) the converted string cannot entirely fit into the output buffer, yielding a string which is not null terminated.</td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>mbstowcs</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td>If the buffer size was specified incorrectly and the attacker controls the input string, then the attacker may be able to induce a buffer overflow and achieve arbitrary code execution.
If measures are not taken to ensure that the result is null-terminated, then an an attacker controlling the input can force an unterminated result. Subsequent operations on the unterminated string may result in the program crashing or in other unexpected behavior.</td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>Whenever mbstowcs() is called.</td><td>The "count" parameter giving the maximum number of characters to be converted must not be larger than the output buffer size measured in characters. Note that the output buffer is for wide characters, so the size in characters will be only half the size in bytes.
The output buffer should be sized to be large enough to hold the converted string. This can be achieved either by (1) attempting the conversion and then enlarging the buffer if necessary, (2) doing a dummy conversion to discover the needed buffer size, allocating a buffer, and then performing the real conversion.
Alternatively, if a truncated result is acceptable, a terminating null should be added to the</td><td>Effective.</td></tr>
</table>
</td></tr>
</table>

	result before it is used. Note that the function mbsrtowcs() may be used to convert the string piecewise if it doesn't make sense to try to convert the entire string at once.
Signature Details	size_t mbstowcs(wchar_t *wcstr, const char *mbstr, size_t count);

Examples of Incorrect Code	`wchar_t destString[20];`
	`const char sourceString[] = "Pretend this string is multi-byte.";`
	
	`// The following has multiple problems:`
	`// 1. The number of characters destString can hold is actually sizeof(destString)/sizeof(destString[0])`
	`// 2. The buffer is not large enough to hold the converted string.`
	`// 3. No check is done to ensure that the entire string was converted.`
	
	`mbstowcs(destString, sourceString, sizeof(destString));`

Examples of Corrected Code	`const char sourceString[] = "Pretend this string is multi-byte.";`
	
	`// Size the output buffer as needed to fit the complete result`
	`int charsToProduce = mbstowcs(NULL, sourceString, 0) + 1; // note error return of -1 is possible`
	`if (charsToProduce == 0) { /* handle error */ }`
	`if (charsToProduce > ULONG_MAX/sizeof(wchar_t)) return error;`

<table>
<tr><td></td><td><pre>wchar_t *destString = (wchar_t
*)malloc(charsToProduce *
sizeof(wchar_t));

mbstowcs(destString, sourceString,
charsToProduce);</pre></td></tr>
<tr><td><b>Source Reference</b></td><td>• ITS4 Source Code Vulnerability Scanning Tool[2]</td></tr>
<tr><td><b>Recommended Resources</b></td><td>• MSDN reference for mbstowcs()[3]
• Linux man page for mbstowcs()[4]</td></tr>
<tr><td><b>Discriminant Set</b></td><td><table>
<tr><td><b>Operating System</b></td><td>• Any</td></tr>
<tr><td><b>Languages</b></td><td>• C
• C++</td></tr>
</table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com